I/S: A JOURNAL OF LAW AND POLICY FOR THE INFORMATION SOCIETY

RICHARD CLAYTON, STEVEN J. MURDOCH & ROBERT N. M. WATSON*

Ignoring the Great Firewall of China

Abstract: The so-called "Great Firewall of China" operates, in part, by inspecting Transmission Control Protocol (TCP) packets for keywords that are to be blocked. If the keyword is present, TCP reset packets are sent to both endpoints of the connection, which then close. However, the original packets pass through the firewall unscathed. Therefore, if the endpoints completely ignore the firewall's resets, the connection will proceed unhindered and the firewall will be ineffective. Once one connection has been blocked, the firewall makes further easy-to-evade attempts to block any more connections from the same machine. This latter behaviour of the firewall can be leveraged into a denial-of-service attack on third-party machines.

---

* University of Cambridge, Computer Laboratory, William Gates Building,15 JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom. Each author can be contacted at: {richard.clayton, steven.murdoch, robert.watson}@cl.cam.ac.uk.

We wish to acknowledge the assistance of a Chinese national we will not name (and who was entirely unaware of the nature of our experiment, and whose webpages contain no illicit material) in providing an extremely convincing practical demonstration of a theoretical idea. Richard Clayton is currently working on the spamHINTS project, funded by Intel Research. Steven J. Murdoch is funded by the OpenNet Initiative.

I. INTRODUCTION

The People's Republic of China operates an Internet filtering system which is widely considered to be one of the most sophisticated in the world.[1] It works, in part, by inspecting web traffic to determine if specific keywords are present.[2] These keywords relate to a variety of matters including groups that the Chinese Government has banned, political ideologies that it considers unacceptable, and historical events that the regime does not wish to have discussed.[3]

It is straightforward to determine that the keyword-based blocking is occurring within the routers that handle the connections between China and the rest of the world.[4] These routers use devices based upon intrusion detection system (IDS) technology to determine whether the content of packets matches the Chinese Government's filtering rules.[5] If a connection from a client to a web server is to be blocked, the router will inject forged TCP resets into the data streams so that the endpoints will abandon the connection.[6] Once blocking has begun, it will remain in place for many minutes and further attempts by the same client to fetch material from the same website will immediately be disallowed by the injection of further forged resets.

In Section 2 of this paper, we discuss the methods available to countries that wish to prevent their citizens from accessing particular Internet content and the strengths and weaknesses of each that have been identified by previous investigators. In Section 3, we present the packet traces we obtained from each endpoint of connections that were blocked by the Chinese firewall system. In Section 4, we propose a

---

[1] OpenNet Initiative, "Internet Filtering in China in 2004–2005: A Country Study," *OpenNet Initiative,* http://www.opennetinitiative.net/studies/china/ONI_China_Country_Study.pdf (accessed October 21, 2007).

[2] OpenNet Initiative, "Probing Chinese Search Engine Filtering," *OpenNet Initiative: Bulletin 005*, http://www.opennetinitiative.net/bulletins/005/ (accessed October 15, 2007).

[3] Ronald J. Deibert and others, eds., *Access Denied: The Practice and Policy of Global Internet Filtering* (Cambridge: MIT Press, 2007).

[4] Nart Villeneuve, "Censorship is in the Router," June 3, 2005, http://ice.citizenlab.org/?p=113 (accessed October 15, 2007).

[5] OpenNet Initiative, "Probing Chinese Search Engine Filtering."

[6] TCP protocol packets with the RST flag bit set. These packets signal that a participant wants the current connection to be immediately closed down and no further traffic transmitted.

model for the operation of this firewall that explains the results we obtained. Then, in Section 5, we show that by ignoring the TCP resets issued by the firewall, we are able to successfully transfer material that was supposed to be blocked and discuss why this method of subversion may prove difficult for the firewall operators to address. In Section 6, we show how the blocking action of the firewall can be leveraged into a denial-of-service attack on third party machines. Finally, in Section 7, we discuss the merits and demerits of this method of evading censorship, consider how websites outside of China might make their material easier to access despite such blocking, and ask to what extent public policy should encourage this.

II. CONTENT BLOCKING SYSTEMS

Three distinct methods of content blocking – packet dropping, Domain Name System (DNS) poisoning and content inspection – have been identified in previous papers by Dornseif,[7] who studied the blocking of right-wing and Nazi material in Nordrhein-Westfalen, and Clayton,[8] who studied the hybrid blocking system deployed by British Telecom (BT) in the United Kingdom to block access to paedophile websites.

A. PACKET DROPPING SCHEMES

In a packet dropping scheme, all traffic to specific Internet Protocol (IP) addresses is discarded and the content hosted there becomes inaccessible. This scheme is low cost and easy to deploy – firewalls and routers offer the necessary features as standard.

Packet dropping schemes suffer from two main problems. First, the list of IP addresses must be kept up-to-date, which could pose some difficulties if the content provider wishes to make it hard for an Internet Service Provider (ISP) to block their websites.[9] Second, the

---

[7] See Maximillian Dornseif, "Government Mandated Blocking of Foreign Web Content," *Security, E-Learning, E-Services: Proceedings of the 17 DFN-Arbeitstagung über Kommunikationsnetze*, eds. Jan van Knop, Wilhelm Haverkamp, Eike Jessen, 617–646 (Dusseldorf, Germany: GI, 2004).

[8] Richard Clayton, "Failures in a Hybrid Content Blocking System," in *Privacy Enhancing Technologies: 5th International Workshop Cavtat, Croatia, May 30-June 1, 2005* (Berlin, Germany: Springer, 2006): 78–92.

[9] Richard Clayton, "Anonymity and Traceability in Cyberspace," Technical Report (2005), http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-653.pdf (for details of the complexity,

system can suffer from "overblocking," that is, all of the other websites that share the same IP address will also be blocked. Edelman investigated the potential extent of "overblocking" and found that 69.8% of the websites for *.com*, *.org* and *.net* domains shared an IP address with fifty or more other websites.[10] Although some of these domain names will have merely been "parked" and are providing a generic webpage, the detailed figures show a continuum of differing numbers of websites per IP address, this reflects the prevailing commercial practice of hosting as many websites as possible on every physical machine.

B. DOMAIN NAME SYSTEM POISONING SCHEMES

A Domain Name System (DNS) poisoning scheme works by failing to provide the correct answer to DNS lookups. When the scheme is in place, any time the system is consulted to translate a textual hostname into a numeric IP address, either no answer is returned or an incorrect answer is given that leads the user to a generic site that serves a warning about accessing forbidden content.

These schemes do not suffer from "overblocking" because no other website is affected when access to a specific host is forbidden. However, because DNS lookups are also used when delivering email, it can be difficult to make a poisoning scheme work correctly if all that is to be blocked is a website and email contact should still be permitted. Dornseif demonstrated that all of the ISPs in his sample had made at least one mistake when implementing DNS poisoning.[11]

C. CONTENT INSPECTION SCHEMES

Most content inspection schemes work by arranging for all traffic to pass through a proxy. The proxy server filters the content by refusing to return results containing forbidden material. These systems can be made extremely precise, potentially blocking single webpages or single images while permitting everything else to pass through unhindered.

---

see the extensive discussion in "Anonymity and Traceability in Cyberspace") (accessed October 15, 2007).

[10] Benjamin Edelman, "Web Sites Sharing IP Addresses: Prevalence and Significance," *Berkman Center for Internet & Society*, http://cyber.law.harvard.edu/people/edelman/ip-sharing (accessed October 15, 2007).

[11] Dornseif, "Government Mandated Blocking," 626–27.

The reason that proxy-based systems are not universally employed is that a system that can cope with the traffic volumes of a major network – or an entire country – would be extremely expensive. In the United States, a Pennsylvania statute requiring the blocking of sites adjudged to contain child pornography was struck down as unconstitutional in September 2004.[12] For cost reasons, the Pennsylvanian ISPs had been using a mixture of packet dropping and DNS poisoning. The overblocking result and "prior restraint" were significant factors in the court's decision.

Nevertheless, proxy-based systems have been deployed both in countries including Saudi Arabia,[13] and Burma,[14] and on specific network providers such as Telenor in Norway.[15] The UK-based BT system studied by Clayton was a hybrid design which utilized a low-cost proxy because only the packets destined for relevant IP addresses would be passed to it. Unfortunately, this permits users to "reverse-engineer" the list of blocked sites. Since these sites provide illegal images of children, this runs counter to the public policy aim of the system.

An alternative method of performing content inspection uses components from an Intrusion Detection System (IDS). The IDS equipment inspects the traffic as it passes by and determines whether or not the content is acceptable. When the content is to be blocked, it will arrange for packets to be discarded at a nearby firewall, or, in the case of the Chinese system, it will issue TCP reset packets which will cause the offending connection to be closed.

An IDS-based system is significantly more flexible than the other schemes and it is much less simple to circumvent. Both Dornseif and Clayton have extensive discussions on how to circumvent the different types of content blocking they identify.[16] However, the IDS approach

---

[12] *Center for Democracy & Technology v. Pappert*, 337 F. Supp. 2d 606 (E.D. Penn. 2004).

[13] King Abdulaziz City for Science and Technology: Local Content Filtering Procedure. Internet Services Unit (2004), http://www.isu.net.sa/saudi-internet/contenet-filtring/filtring-mechanism.htm (accessed October 15, 2007).

[14] OpenNet Initiative, "Internet Filtering in Burma in 2005: A Country Study," *OpenNet Initiative*, http://www.opennetinitiative.net/burma/ONI_Burma_Country_Study.pdf (accessed October 15, 2007).

[15] Telenor, "Telenor and KRIPOS Introduce Internet Child Pornography Filter," press release, September 21, 2004, http://presse.telenor.no/PR/200409/961319_5.html (accessed October 15, 2007).

[16] Dornseif, "Government Mandated Blocking," 642-44; Clayton, "Failures in a Hybrid Content Blocking System," 78–92.

ought to be able to detect the traffic no matter what evasion scheme is tried, provided that the traffic remains in the clear and is not encrypted or obfuscated in a manner that the IDS is unable to convert to a canonical format before coming to a decision.[17]

III. HOW THE CHINESE FIREWALL BLOCKS CONNECTIONS

In our experiments, we were accessing a website based in China (within the Chinese firewall) from several machines based in Cambridge, England (outside the Chinese firewall). The Chinese firewall system, as currently deployed, is known to work entirely symmetrically.[18] It detects content to be filtered as it passes in both directions.[19] By issuing all the commands from the Cambridge end we avoided any possibility of infringing Chinese law.

We started by accessing a Chinese webpage in a normal fashion and recording the packets that flowed between the endpoints. We then issued a request that intentionally triggered the censorship action of the "Great Firewall" and observed how it used reset packets to cause the connection to close. We repeated the "normal" request (without the triggering phrase) and found, rather unexpectedly, that this connection was also blocked. The rest of this section gives the detailed results of what we observed.

A. BLOCKING WITH RESETS

Initially we accessed a simple webpage, which arrived in an entirely normal manner, just as would be expected. As can be seen from the details of the packets we present below, after the initial TCP three-way handshake (SYN,[20] SYN/ACK,[21] ACK)[22] the client (using

---

[17] The IDS can deal with variations in what it is looking for by converting irrelevant representational details in what it observes into a standardized form. It can then compare this standard version with its list of what is to be blocked (also held in a canonical form) and hence make the correct decision.

[18] This symmetry is necessarily present because it permits the firewall to block both requests that are deemed to be unacceptable and the return of unacceptable content.

[19] Villeneuve, "Censorship is in the Router."

[20] The SYN (synchronise) flag is set to distinguish the first packet sent when a TCP connection is being opened.

port 53382 in this instance) issues a Hypertext Transfer Protocol (HTTP) GET command to the server's http port (tcp/80) for the top level page (/), which is then transferred normally. We were using the Netcat program (nc) to issue the request, rather than a standard web browser, so that we might avoid extraneous detail. The packet traces were captured by Ethereal, a network protocol analyzer, but we present them in a generic format.

cam(53382) → china(http) [SYN]
china(http) → cam(53382) [SYN, ACK]
cam(53382) → china(http) [ACK]
cam(53382) → china(http) GET / HTTP/1.0<cr><lf><cr><lf>
china(http) → cam(53382) HTTP/1.1 200 OK
(text/html)<cr><lf> *etc...*
china(http) → cam(53382) *... more of the webpage*
cam(53382) → china(http) [ACK]
... and so on until the page was complete

We then issued a request which included a small fragment of text that we expected to cause the connection to be blocked and this occurred promptly:

cam(54190) → china(http) [SYN]
china(http) → cam(54190) [SYN, ACK] TTL=39
cam(54190) → china(http) [ACK]
cam(54190) → china(http) GET /?falun
HTTP/1.0<cr><lf><cr><lf>
china(http) → cam(54190) [RST] TTL=47, seq=1, ack=1
china(http) → cam(54190) [RST] TTL=47, seq=1461, ack=1
china(http) → cam(54190) [RST] TTL=47, seq=4381, ack=1
china(http) → cam(54190) HTTP/1.1 200 OK

---

[21] The response to the SYN packet has both the SYN and ACK (acknowledge) flags set, which uniquely distinguishes this second "SYN/ACK" packet in the TCP connection opening procedure.

[22] The precise details of the TCP protocol and the details of how (and why) it initiates a connection by swapping three packets with the SYN, SYN/ACK and ACK flags set, respectively, can be found in any good communications networking textbook, such as W. Richard Stevens, *TCP/IP Illustrated, Volume 1, The Protocols* (Reading, MA: Addison-Wesley, 1994).

(text/html)<cr><lf> *etc...*
cam(54190) → china(http) [RST] TTL=64, seq=25, ack zeroed
china(http) → cam(54190) *... more of the webpage*
cam(54190) → china(http) [RST] TTL=64, seq=25, ack zeroed
china(http) → cam(54190) [RST] TTL=47, seq=2921, ack=25

The first three reset packets had sequence values that corresponded to the sequence number at the start of the GET packet; that value plus 1460, and that value plus 4380 (3 × 1460).[23] We believe that the firewall sends three different values to try and ensure that the reset is accepted by the sender, even if the sender has already received ACKs for "full-size" (1460 byte) packets from the destination. Setting the sequence value of the reset packet "correctly" is necessary because many implementations of TCP/IP now apply strict checks that the value is within the expected "window."[24] The security vulnerabilities inherent in failing to check for a valid sequence value were first pointed out by Watson in 2004.[25]

The trace also shows part of the webpage arriving from the Chinese machine after the connection had already been aborted. The Cambridge machine therefore sent its own TCP resets in response to these two (now) unexpected packets. Note that it placed a zero into the acknowledgement fields rather than using a value relative to the randomly chosen initial value.

All of the reset packets arrived with a time-to-live (TTL) field value of forty-seven[26] whereas the packets from the Chinese webserver

---

[23] When we arranged for the endpoints to use the TCP timestamp option and the packets contained an extra 12 bytes of TCP options, we observed that these values changed to multiples of 1448.

[24] TCP labels all data packets with a sequence number to indicate which chunk of the transmitted data each contains. When packets are lost, delayed or even duplicated, the sequence number permits the data stream to be reliably reconstructed. The "window" is the largest amount of data that can be sent without an acknowledgement being received. On today's Internet, checking that sequence numbers lie within the window (and that a reset packet contains an expected sequence number) is an important security measure that prevents third parties from disrupting a connection.

[25] Paul A. Watson, "Slipping in the Window: TCP Reset Attacks," *Open Source Vulnerability Database*, http://osvdb.org/reference/SlippingInTheWindow_v1.0.doc (accessed October 15, 2007).

[26] The time-to-live value (TTL) is initialized by the sender of a packet and its value is decremented by every router that the packet passes through. The idea is to ensure that packets cannot endlessly circulate because when the count reaches zero the packet is discarded.

always had a TTL value of thirty-nine, indicating that they were from a different source. If both sources set an initial value of sixty-four, then this would indicate the resets were generated eight hops away from the webserver. We used the **traceroute** program to determine the route that the packets follow; the program shows that the second router is situated within the China Netcom Corporation network (AS9929) after the traffic is passed across from the Sprint network (AS1239).

We also examined this blocked connection from the point of view of the Chinese webserver:

cam(54190) → china(http) [SYN] TTL=42
china(http) → cam(54190) [SYN, ACK]
cam(54190) → china(http) [ACK] TTL=42
cam(54190) → china(http) GET /?falun HTTP/1.0<cr><lf><cr><lf>
china(http) → cam(54190) HTTP/1.1 200 OK (text/html)<cr><lf> *etc...*
china(http) → cam(54190) *... more of the webpage*
cam(54190) → china(http) [RST] TTL=61, seq=25, ack=1
cam(54190) → china(http) [RST] TTL=61, seq=1485, ack=1
cam(54190) → china(http) [RST] TTL=61, seq=4405, ack=1
cam(54190) → china(http) [RST] TTL=61, seq=25, ack=1
cam(54190) → china(http) [RST] TTL=61, seq=25, ack=2921
cam(54190) → china(http) [RST] TTL=42, seq=25, ack zeroed
cam(54190) → china(http) [RST] TTL=42, seq=25, ack zeroed

As can be seen, when the "bad" packet was detected the firewall also sent resets, the "[RST]" packets, to the Chinese machine, but these resets arrived after the GET packet (and after the response had commenced). The last two resets (with zeroed ACK values) were the ones that were sent by the Cambridge machine.

The other resets (generated because **falun** was present) arrived at the Chinese webserver with a TTL value of sixty-one, which is consistent with them being generated three hops away with an initial count of sixty-four. This differs from the eight-hop offset we observed from Cambridge. However, it is possible that there is more than one

---

Inspecting the TTL value can therefore be used to deduce the length of the path a packet has travelled from its origin.

device that is generating resets – or the initial count may have been adjusted to be different from sixty-four. We do not currently have a definitive explanation for this lack of symmetry in the TTL values for the reset packets.

The first three blocking resets were also set to a range (+25, +1485, +4405) of sequence numbers in an attempt to ensure that at least one was accepted and in fact the +25 packet will have reset the connection.[27] Examining the acknowledgement values in the fourth and fifth resets received indicates that they are the responses to the two packets that the server managed to send before the connection was reset.

B. IMMEDIATE RESET OF CONNECTIONS

The Chinese Firewall does not just inspect content; it has other blocking rules as well. Having made a "bad" connection, we found that for a short period, all web traffic between the same two hosts was blocked. This blocking occurred before any determination could possibly have been made as to the content. This can also be seen in the previous example – but it can be seen to apply to new connections as well. For example, immediately after the example documented above we saw this:

cam(54191) → china(http) [SYN]
china(http) → cam(54191) [SYN, ACK] TTL=41
cam(54191) → china(http) [ACK]
china(http) → cam(54191) [RST] TTL=49, seq=1

Here the reset packet came from the firewall (which sent a reset to the webserver as well) and the client closed. However, there is a race condition here, if the client manages to send out its GET packet in the short time period before the reset arrives from the firewall, multiple further resets will arrive from the firewall (even if the GET is entirely innocuous). There are further resets as well from the webserver. This occurs because the webserver receives its own reset from the firewall and tears down the connection before the GET arrives. Since the GET

[27] If the resets had arrived *before* the GET packet, then the resets would *not* have been accepted. The server is using the FreeBSD operating system and during this stage of a connection its TCP stack will only accept a reset when the sequence number exactly matches the last acknowledgement sent. This behavior is intended to provide protection against denial-of-service attacks. Before the GET arrives that value is +1 hence all of the resets would be ineffective.

is no longer associated with an open connection, the webserver follows the protocol and sends back a reset in response.

It should be noted that the firewall does not attempt to reset the connection at the SYN stage (the first stage of the three-way handshake) but waits for the SYN/ACK (the second packet). Although the client could immediately be sent valid reset packets when the SYN is seen, it is only when the SYN/ACK packet is observed that a reset can be constructed with valid values for the server to act upon.[28]

In our experiments, we found that the length of time for which a pair of endpoints would be prevented from communicating was somewhat variable. Sometimes the blocking would only last for a few minutes yet at other times the block would be present for most of an hour. The average value was around twenty minutes, but because we saw significant clustering of times around specific values, we suspect that different firewall system components may be setting different time delays. A better understanding of which firewall component was to handle our traffic would enable us to predict the blocking period fairly accurately.

C. Application to Other Chinese Networks

We obtained a list of Chinese Autonomous Systems (ASs) and from it generated a list of all Chinese subnets that were present in the global routing table.[29] We then used a modified **tcptraceroute** to determine which ASs were handling traffic as it crossed from international networks into China, and from this learned the identities of the major Chinese border networks. These turned out to be: AS4134, AS4837, AS7497, AS9800, AS9808, AS9929, AS17622, AS24301 and AS24489. We then selected an example webserver within each of these ASs and found that all of these networks (except AS24489: Trans-Eurasia Information Network) exhibited similar reset behavior to that described in detail above. From this we conclude that, while our results are extremely typical of the "Great Firewall of

---

[28] The SYN/ACK packet contains the sequence numbers selected by *both* ends of the connection.

[29] An Autonomous System (AS) is a major network that is owned by a particular ISP. The list we used was CERNET's "China ASN List," http://bgpview.6test.edu.cn/bgp-view/cur_ana/ipv4cn/china_asnlist.shtml (accessed October 15, 2007) (Internet routers hold a list of which blocks of address space (subnets) can best be reached through each of their connections to other routers at other ISPs. This "global routing table" is expressed in terms of ownership of addresses by particular ASs).

China" as it existed in late May 2006, they are not necessarily universally applicable.[30]

IV. DESIGN OF THE CHINESE FIREWALL

Based on the results of our experiments and descriptions of the type of devices and technologies known to be employed in China – such as Cisco's "Secure Intrusion Detection System"[31] – we propose the following model for the operation of a router that is a part of the Chinese firewall. This model fits our observations well, but it remains speculative because Chinese network providers do not publish any of the specifications for their systems.

When a packet arrives at the router, it is immediately placed into an appropriate queue for onward transmission. The packets are also passed to an out-of-band IDS device within which their content is inspected. If the packet is considered to be "bad" by the IDS device (because of a keyword match) then three TCP reset packets – with the three different sequence numbers – are generated for each endpoint and given to the router to be transmitted to their destinations.[32]

The IDS is a logically separate device; it would be extremely complicated to give it the capability of removing "bad" packets from the router transmission queue or to delay them while a decision is being made. However, it is relatively simple to permit the IDS to issue resets and thereby cause connections to close.

If there is some congestion within the router and the IDS device is keeping up, then the reset packet will be sent ahead of the "bad" packet; this is what we mainly observed in our experiments, although sometimes the reset packet would lag behind. The values chosen for the reset packets strongly suggest that the designers were concerned that if there is some congestion within the IDS device, compared with the router, then several "bad" packets may have already been transmitted and so the reset packets will reach the destination after these have arrived. Note that in the design we are describing, if the

---

[30] See Jedidiah R. Crandall and others, "ConceptDoppler: A Weather Tracker for Internet Censorship" (14th ACM Conference on Computer and Communications Security, Alexandria, VA, October 29–November 2, 2007) http://www.cs.unm.edu/~crandall/concept_doppler_ccs07.pdf (accessed October 15, 2007).

[31] Earl Carter, *Secure Intrusion Detection Systems* (Indianapolis: Cisco Press, 2001).

[32] i.e., Equipment that can inspect the content of packets that is "off to one side" of the actual connection so that it can detect "bad" traffic but cannot directly affect its flow.

designers had not caused these extra resets to be sent, the firewall might not have blocked connections reliably when it became busy.

Once the IDS system has detected behavior it wishes to block, it might have blocked the traffic by adding a simple discard rule to the main router rather than issuing resets.[33] We strongly suspect that this does not scale well within major, high-speed routers, but that scaling the blocking within the IDS systems is cheaper and easier.

We have already observed, from the time periods for which connections were blocked, that there seemed to be several devices providing the firewall functionality. We ran a further experiment and sent 256 packets containing the offending string through the firewall. Although these packets came from a single machine, we set their source addresses to 256 consecutive IP address values, *viz.*: the Chinese firewall would believe that 256 different, albeit related, machines were sending content that was to be blocked. We observed that the reset packets that were returned to us would sometimes arrive "out of order."

The modern Internet generally arranges for packets to be processed in FIFO (first-in, first-out) queues,[34] so the simplest explanation for the lack of ordering was that different packets had been passed to different IDS systems whose own FIFO queues were not equally loaded at the moment they issued the resets. Unfortunately, we found that the experiment engendered so much packet loss (not all of the resets were returned for all of the connections) that it was not possible to form a view as to how far out of order packets could come. For this reason we were unable (by modeling the queues) to establish a lower bound on the number of parallel IDS devices. We intend to return to this experiment at a later time.

A. FIREWALL "STATE"

There is no evidence that the out-of-band IDS devices communicate with each other to create a shared notion of the "state" of connections that pass through the firewall. Experiments demonstrate that triggering a firewall in one border network did not affect the traffic passing through another.

Even where "state" might be expected to be preserved – within the IDS devices – there is no stateful TCP inspection – *viz.*: the devices

---

[33] Routers often have the facility to discard packets that match particular criteria.

[34] Yi Wang, Guohan Lu, and Xing Li, "A Study of Internet Packet Reordering," *Information Networking* (Heidelberg, Germany: Springer-Berlin, 2004): 350–359.

considering each packet on its individual merits – so that arranging for the **?falun** query to be split between two adjacent packets is sufficient to avoid detection. Furthermore, the devices are unaware of whether an open connection exists, so for many of our tests, we did not perform the three-way handshake to open a connection. Instead, we simply sent the packet containing the HTTP GET request. In fact, apart from the ongoing blocking of traffic after the initial detection occurs, there is no evidence for the IDS devices doing anything other than acting upon one packet at a time.

V. DELIBERATELY IGNORING RESETS

The firewall relies entirely upon the endpoints implementing the TCP protocol[35] in a standards-compliant manner, which means that they will abort the connection when a reset packet is received. The firewall could sometimes be slightly caught out, as we noted above, when the resets beat the GET packet to the destination. In that instance, the resets were ignored by the careful validation that was applied. Nevertheless, the connection was successfully torn down as soon as the next packet transited the firewall; hence this did not make much overall difference.

Now consider what happens if the endpoints do not conform to the standards and the TCP resets are entirely ignored. We might expect the firewall to have no impact on HTTP transfers, despite the IDS system having been triggered.

We therefore conducted a further experiment with both of the endpoints ignoring TCP resets. We could have achieved this in a number of different ways but we chose to set appropriate rules within packet filtering firewalls. Within Linux, we installed **iptables** and gave the command:

iptables -A INPUT -p tcp --tcp-flags RST RST -j DROP

which specifies that incoming TCP packets with the RST flag set are to be discarded. If we had been using FreeBSD's **ipfw** the command would have been:

ipfw add 1000 drop tcp from any to me tcpflags rst in

---

[35] J. Postel, ed., "Transmission Control Protocol, DARPA Internet Program Protocol Specification" (memo, Network Working Group Request for Comments, September 1981) http://www.ietf.org/rfc/rfc793.txt (accessed October 21, 2007).

Once we were discarding TCP resets, we found that we could indeed transfer a webpage without any blocking occurring. Examining the traffic at the Cambridge end of the connection, we saw the following results:

cam(55817) → china(http) [SYN]
china(http) → cam(55817) [SYN, ACK] TTL=41
cam(55817) → china(http) [ACK]
cam(55817) → china(http) GET /?falun HTTP/1.0<cr><lf><cr><lf>
china(http) → cam(55817) [RST] TTL=49, seq=1
china(http) → cam(55817) [RST] TTL=49, seq=1
china(http) → cam(55817) [RST] TTL=49, seq=1
china(http) → cam(55817) HTTP/1.1 200 OK (text/html)<cr><lf> *etc...*
china(http) → cam(55817) *... more of the webpage*
cam(55817) → china(http) [ACK] seq=25, ack=2921
china(http) → cam(55817) *... more of the webpage*
china(http) → cam(55817) [RST] TTL=49, seq=1461
china(http) → cam(55817) [RST] TTL=49, seq=2921
china(http) → cam(55817) [RST] TTL=49, seq=4381
cam(55817) → china(http) [ACK] seq=25, ack=4381
china(http) → cam(55817) [RST] TTL=49, seq=2921
china(http) → cam(55817) *... more of the webpage*
china(http) → cam(55817) *... more of the webpage*
cam(55817) → china(http) [ACK] seq=25, ack=7301
china(http) → cam(55817) [RST] TTL=49, seq=5841
china(http) → cam(55817) [RST] TTL=49, seq=7301
china(http) → cam(55817) [RST] TTL=49, seq=4381
china(http) → cam(55817) *... more of the webpage*
china(http) → cam(55817) [RST] TTL=49, seq=8761
... and so on until the page was complete

viz.: the webpage was transferred in a normal manner except for the TCP reset packets generated by the firewall (marked with [RST]s in the results). Since these were all ignored (there were twenty-eight resets sent in total) they had no effect on the client's TCP/IP stack which continued to accept the incoming webpage and it can be seen to

be issuing ACK packets as appropriate. A similar pattern of RSTs mixed in amongst the real traffic could also be seen at the Chinese end of the connection.

Hence, by simply ignoring the packets sent by the "Great Firewall," we made it entirely ineffective! This will doubtlessly disappoint its implementers.

A. BLOCKING WITH CONFUSION

As well as blocking further connections by issuing TCP resets once the connection was established we observed that parts of the firewall occasionally used an additional strategy. On some pairs of endpoints (apparently at random) we saw a forged SYN/ACK packet arrive from the firewall. This packet was pretending to be the second packet of the three-way handshake but it contained an apparently random (and hence invalid) sequence number.

If the SYN/ACK packet generated at the firewall arrives at the client before the real SYN/ACK then the connection fails: the client records the random sequence number from the specious SYN/ACK and so it returns what the server considers to be an incorrect ACK value. This triggers a reset packet from the server that causes the client to close. In practice, when the client is prompt in sending its GET, as in the trace below, a number of other packets are seen and they cause both the firewall and the server to respond with further resets:

cam(38104) → china(http) [SYN]
china(http) → cam(38104) [SYN, ACK] TTL=105
cam(38104) → china(http) [ACK]
cam(38104) → china(http) GET / HTTP/1.0<cr><lf><cr><lf>
china(http) → cam(38104) [RST] TTL=45, seq=1
china(http) → cam(38104) [RST] TTL=45, seq=1
china(http) → cam(38104) [SYN, ACK] TTL=37
cam(38104) → china(http) [RST] TTL=64, seq=1
china(http) → cam(38104) [RST] TTL=49, seq=1
china(http) → cam(38104) [RST] TTL=45, seq=3770952438
china(http) → cam(38104) [RST] TTL=45, seq=1
china(http) → cam(38104) [RST] TTL=45, seq=1
china(http) → cam(38104) [RST] TTL=37, seq=1
china(http) → cam(38104) [RST] TTL=37, seq=1

Dealing with this new firewall strategy is more difficult than dealing with the forged reset packets. The problem is that even if the client ignores the (entirely valid) reset from the server, it continues to have an incorrect understanding of the server's sequence number and cannot "synchronize" with the server to complete the three-way handshake and connect.

Of course if, as occasionally happens, the specious SYN/ACK from the firewall arrives after the SYN/ACK from the webserver, then it will be ignored by the client and will not cause any confusion. The firewall still attempts to tear down the connection with forged reset packets but, just as before, ignoring these resets means that a blocked webpage can still be viewed.

Deciding which of the two incoming SYN/ACK packets is genuine is clearly essential. In the examples we saw, they were easy to distinguish. The firewall version had various distinctive features such as a distinctive TTL value, a lack of a do-not-fragment (DF) flag, and no TCP options. Forged SYN/ACK packets are therefore, at present, just as easy to filter as resets and the Chinese firewall is once again ineffective. Moreover, this strategy is only used once an attempt has been made to block a previous connection. Hence the expected TTL value for the server could be remembered by the client whereas the firewall will not know what value to accord its forged packet.

However, with increasing sophistication, the firewall might manage to forge SYN/ACK packets with no detectable differences. The client could simply take the view that the firewall packet was the one arriving first. However, if the firewall countered this by sometimes delaying its SYN/ACK packet then a complex "game" could result with ever more abstruse strategies as the client attempted to guess which reset came from the firewall. It should be noted that, because webpage fetching often involves multiple connections, the firewall operators might feel that they had "won" the game by blocking a proportion of access attempts rather than all of them.

An effective client strategy, if both the client and the server are discarding resets, would be to arrange to treat all incoming SYN/ACK packets (the firewall might in the future send more than one) as valid. The client should then record their sequence values and ACK all of them. The client must then continue to consider all values potentially correct and keep track of all the possibilities, until it receives an ACK from the server that permits it to confirm which value is actually correct. However, this strategy would be somewhat complex and is well beyond the capabilities of simple packet-filtering systems such as **iptables** or **ipfw**.

A further round of this new "game" would be for the firewall to forge an ACK for all of the client's packets. It should be possible for

the client to see through this subterfuge by discarding values for which a genuine-looking RST is received from the server, so the firewall would need to forge these. Once again the strategies may become arbitrarily complex. The endpoints do have an advantage in that they can eventually conclude whether packets are being generated by the other (stateful) endpoint or by a stateless firewall. However, should the firewall start to keep "state," this major architectural change (albeit almost certainly at significant cost) would open up many other strategies and the advantage would swing decisively to the firewall.

Unfortunately, it must be noted that firewall generated SYN/ACK packets cannot be securely dealt with by a change to the TCP/IP stack at the server end of the connection. The server is entirely able to work out that the client is continually responding with the "wrong" ACK value and so it could negate the effect of the interference by retrospectively altering its own state to correspond with the value from the forged SYN/ACK packet. However, doing this would remove an important security procedure documented by Bellovin and would therefore allow access by malicious systems that forged source IP addresses so as to pretend to be another machine.[36]

Making secure connections in the presence of adversaries that can "sniff" packets and add forged packets of their own has, of course, been well studied in the context of cryptographic key exchange protocols. The open question is to what extent fairly simple modifications to existing TCP/IP stacks will continue to be sufficient to overcome the strategies available to the Chinese firewall operators, given the architectural limitations of their current design.

VI. DENIAL-OF-SERVICE ATTACKS

As we have already noted, a single TCP packet containing a request such as **?falun** is sufficient to trigger blocking between the destination address and source address for periods of up to an hour. If the source of the packet is forged, this permits a (somewhat limited) denial-of-service attack, which will prevent a particular pair of endpoints from communicating. Depending upon their motives, this might be sufficient for some attackers. For example, it might be possible to identify the machines used by regional government offices and prevent them from accessing "Windows Update," or prevent a particular ministry from accessing specific United Nations websites, or

---

[36] S. Bellovin, memorandum, May 1996, in *Network Working Group Request for Comments*, "Defending Against Sequence Number Attacks," http://www.ietf.org/rfc/rfc1948.txt (accessed October 15, 2007).

prevent access by Chinese embassies abroad to particular Chinese websites "back home."

Our calculations suggest that the denial-of-service could be reasonably effective even if operated by a lone individual on a dial-up connection. Such an individual could generate approximately 100 triggering packets per second, and hence – assuming that blocking was in place for the average period of twenty minutes – some 120,000 pairs of end-points could be permanently prevented from communicating.

Of course, current denial-of-service attacks are seldom instantiated by single dial-up machines but by large numbers of machines on much faster connections. Hence the 120,000 value can be multiplied to taste. However, it may well be that the IDS components of the firewall do not have the ability to record substantial numbers of blocked connections – so the actual impact is likely to be limited by this type of resource consideration. It should also be noted that while the IDS is handling an attempted denial-of-service attack it will have fewer resources to devote to recording information about other connections thereby temporarily reducing its effectiveness.

A. LIMITATIONS ON THE DENIAL-OF-SERVICE ATTACK

Further experiments showed that the firewall's blocking was somewhat more complex than we have explained so far; hence a denial-of-service attack would not necessarily be quite as effective as it initially seemed.

First, the blocking is only applied to further connections with similar port numbers.[37] The algorithm being used by the firewall only blocks the 128 TCP port numbers where the most significant nine bits of their value are identical to those of the port number of the connection that triggered the blocking. For a system such as Windows that allocates the ephemeral port numbers sequentially, this would mean that an average of sixty-four further connections would be blocked (therefore, occasionally, if a port number such as 4095 – whose least significant seven bits are 1111111 – was used in a triggering connection, there would be no further blocking). Conversely, on a system such as OpenBSD, which uses ephemeral port

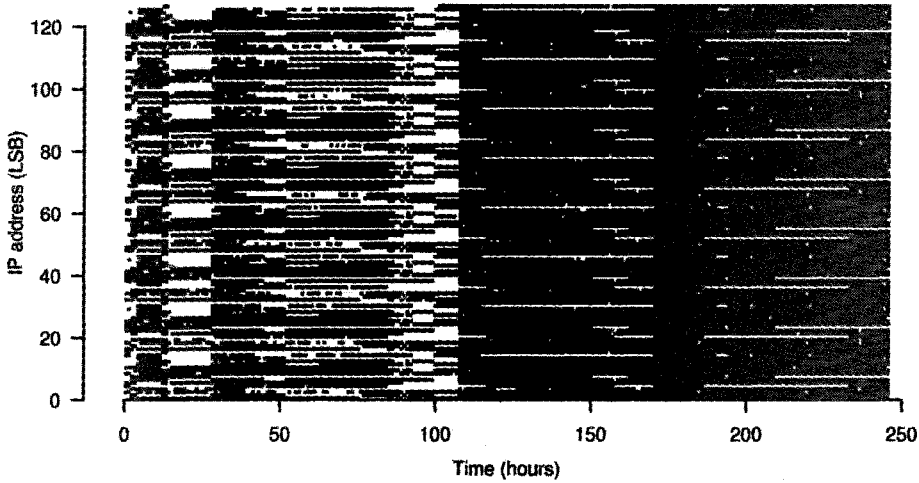
[37] Each connection made to a web server will use the "well-known" port 80 at the server end but the client will allocate a new "ephemeral" port number for each connection. This allows multiple connections to be made to the same server in parallel with the port number distinguishing them. The ephemeral port number values have no particular significance and some systems allocate them randomly, whereas others work through values 1025, 1026, 1027 . . . sequentially.

numbers pseudo-randomly, the chance of another connection being blocked is only about 1 in 500.

We do not have a definitive explanation as to why the firewall behaves this way. It would seem much simpler and more effective to just block every connection to the same endpoints without worrying about the port number.[38] It is possible that the aim is to avoid penalizing other users of Network Address Translation (NAT) devices when just one user has been blocked or it may be that the port number helps determine which particular IDS machine is given the packet. However, it may just be that the behaviour is meant to appear mysterious and therefore more menacing.

From the point of view of a denial-of-service attacker, the consequence is that the firewall must be persuaded to block all possible port number ranges. So, unless there are special circumstances that allow the attacker to guess which ephemeral port numbers will be used in the near future, there will be an increase, by a factor of about 500, in the number of packets that must be sent to ensure that a machine is blocked.

Figure 1: Blocking of "bad" strings by the Chinese firewall.



We tested from 256 adjacent IP addresses once an hour for ten days in early February 2006. Results for the first 128 are shown; the pattern was very similar for the others. The dark blobs indicate that

[38] Web traffic was blocked not only on tcp/http port 80, but also on other port numbers. However, only a single server port was ever blocked – no adjacent ports were affected – nor was tcp/https (port 443) blocked when port 80 was.

the access was blocked; the white indicate where there was no blocking. When the result was indeterminate (no response at all) the color is a mid-gray. An obvious change in firewall configuration (to block more IP addresses) is visible after 110 hours.

Second, not all IP addresses had their traffic inspected. Every hour we sent a rapid burst of requests containing **?falun**, one packet from each of a block of 256 consecutive IP addresses. Initially, about two-thirds of each set of packets were blocked, with the address selection varying over time. However, after a few days, almost all packets caused blocking behaviour. We were unable to reverse-engineer the algorithm that determined which IP addresses had their packets scanned, although distinctive patterns within the IP address selections strongly suggest by their regularity that quite a simple mechanism has been deployed.[39] The most likely explanation for the failure to block every request is a lack of resources; two-thirds of the traffic may be all that the content scanning system can handle. Clearly, if a proportion of machines are being excused packet inspection at a particular time, then at that time it will not be possible to mount a denial-of-service attack on them.

Finally, we observe that these experiments, as is the case with all the experiments we made, were performed using a small number of endpoints both outside and within China. Although we saw reasonably consistent results, with a system as complex as the "Great Firewall of China," it is entirely possible that we failed to observe significant aspects of its behaviour. Hence, although we believe that a denial-of-service attack may succeed in many circumstances, we cannot say that an attack on an arbitrary pair of endpoints would succeed.

VII. STRATEGIC CONSIDERATIONS

In order for traffic to pass unhindered, through the Chinese firewall, to "protected" machines it is necessary for both endpoints to ignore resets. Machines in the "rest of the world" that wish to be accessed from China should have no difficulty arranging for a reconfiguration. However, the individual at the Chinese end of the connection may not wish to install special software. Such an individual's difficulty is that the firewall may not only be blocking connections but also logging what it has done and who was involved. This might lead to an investigation resulting in the discovery of the

[39] See Figure 1, page 292.

specially installed software and an unenlightened view might be taken of the motives for installing it.

The packet inspection capabilities of the Chinese firewall can also be evaded by the use of encryption. If the authorities detected encrypted traffic, perhaps by statistical analysis of the content, then the same problem of specially installed software would arise when the endpoint was visited. However, since encryption systems typically discard the keys that they negotiated for a particular connection as soon as the connection closes, it might not be possible to demonstrate that the traffic had been, say, pornography rather than political speech. In the case where the firewall is breached by discarding resets the firewall can make a record of which trigger caused it to attempt to block the connection with this information the authorities could consult their logs and treat the two types of access differently. As a result, some might view discarding reset packets as having an advantage over the use of encryption.

The Chinese authorities might be forced to take a more tolerant view of the use of reset discarding software by their citizens if this was to become universally deployed and the resets were discarded for completely unrelated reasons. So we now turn to a consideration of what these unrelated reasons might be.

Other work on "software firewalls" has shown that TCP resets are routinely discarded with few side effects.[40] The main purpose of a TCP reset is to provide a rapid way of reporting that incoming traffic is unwelcome. However, if the remote machine is well-behaved, then very little more traffic will arrive if the packets are simply ignored, rather than responded to with a reset.

Nevertheless, some people may not wish to discard every TCP reset; an alternative strategy is possible.[41] At present, inspection of the TTL values provides a simple method of distinguishing the resets generated by the firewall from any resets sent by the other end of the connection. In particular, we note that Watson's reset attack, whereby third parties forge resets to close down connections, is usually resisted by careful validation of the sequence numbers of reset packets.[42] Validating the TTL value in the reset packet to ensure that it is similar

---

[40] See Clayton, "Anonymity and Traceability," 81.

[41] In the future the Chinese firewall might block connections with FIN packets, used to mark the normal close of a connection, rather than resets (an abnormal close). Ignoring all FIN packets would upset normal operations and so this alternative strategy would then be the more appropriate.

[42] Watson, "Slipping in the Window."

to the TTL value seen for the rest of the connection would improve the chances of spotting forged resets. One of the present authors has developed a twenty-line patch for FreeBSD[43] that discards resets whose TTL radically differs from other incoming packets on the connection. Experience so far has been very positive. It is unlikely that other operating systems or "personal firewalls" would find it onerous to provide the same facility.

Of course, the Chinese firewall can be adapted to make the proposed method of circumvention harder to achieve. In particular, it could trivially ensure that the TTL value was correct on reset packets sent in the same direction as triggering packets, although getting it correct for resets sent in the other direction would be difficult because Internet routing is often asymmetric, and the firewall cannot be expected to see both directions of traffic.

However, it will continue to be complex to arrange to remove packets from router queues (or even to delay them until a decision on their content has been made). Unless packets can be prevented from reaching their destination, our basic method – ignoring everything the firewall says – will continue to work.

A completely different firewall strategy would be to refuse to route any further packets to sites that have triggered the blocking behavior. However, we have already noted that this may scale very badly because it must be done "in-line" with the fast path through the routers, and of course, full-scale blocking would increase the effectiveness of the denial-of-service attacks we discussed above.

A. PUBLIC POLICY INITIATIVES TO BREACH THE "GREAT FIREWALL"

There has been considerable political interest, particularly in the United States, in the extent to which companies outside of China have been assisting the Chinese government by suppressing information and locating dissidents and bloggers with dissident opinions. In particular, a number of major U.S. corporations were castigated for their policies and actions at a congressional hearing in February 2006.[44] However, this interest in circumventing Chinese filtering technologies goes back much further. For example, SafeWeb, which was partially funded by

---

[43] Robert N. M. Watson, "Patches Associated with My Academic Research," http://www.cl.cam.ac.uk/~rnw24/patches (accessed October 15, 2007).

[44] Suzanne Goldenberg, "Congress Accuses Google of Collusion," *The Guardian*, February 16, 2006, http://www.guardian.co.uk/china/story/0,,1710616,00.html (accessed October 15, 2007).

the CIA, ran an anonymity-providing web proxy between 2000 and 2003. In conjunction with this, SafeWeb developed an anti-censorship technique dubbed TriangleBoy.[45] More recently the Canadian-based Psiphon project launched in late 2006 aims to permit "citizens in uncensored countries to provide unfettered access to the Net through their home computers to friends and family members who live behind firewalls of states that censor."[46]

One might therefore expect there to be considerable interest in the technique described in this paper for circumventing the Chinese firewall by ignoring its reset packets. There is of course the risk of an "arms race," so that ever more complex strategies are required on both sides. Nevertheless, making the firewall ineffective is, at the moment, remarkably straightforward; albeit for the scheme to work, it is necessary for *both* the webserver hosting the content outside of China and the web browser operated inside of China to discard resets. The incentives for webservers to implement reset dropping seem obvious, because they will wish to make their content available within China. However, it becomes far more complicated when one starts to consider the situation inside China and the incentives for making the necessary changes to the web browser (and the rest of the operating system for the personal computer on which it runs). Although all of this software is running on machines inside China, in practice the browser and the operating system were developed in the rest of the world. In particular, the vast majority of the installed software base is versions of Windows, which is written by Microsoft, Inc.

The public policy question that we pose is whether it is appropriate to encourage or compel Microsoft to change their programs to assist in circumventing the Chinese Firewall. There would certainly seem to be wide-ranging condemnation of Chinese censorship, so anti-censorship measures would surely be approved of by both political and public opinion.[47]

As we noted at the beginning of this section, the technical arguments against the change are limited and it might conceivably improve security against third-party attack (for the firewall machines

---

[45] SafeWeb, "TriangleBoy Whitepaper," *SafeWeb*, 2003, http://web.archive.org/web/20030417171335/http://www.safeweb.com/tboy_whitepaper.html (accessed October 15, 2007).

[46] Psiphon, http://psiphon.civisec.org (accessed October 15, 2007).

[47] Stokely Baksh, "US Calls for Fall of Great Firewall," *United Press International*, February 15, 2006; Kate Allen, "Today, Our Chance to Fight a New Hi-Tech Tyranny," *Observer*, May 28, 2006; Cory Doctorow, "See No Evil?," *Guardian*, July 6, 2007.

are just a specific example of a third party interfering with web traffic). However, it is quite possible that Microsoft (and the other operating system and browser companies) would be unwilling to antagonize the Chinese Government and so they might delay making any changes to their products until compelled to do so.

It is commonplace to observe that software is easy to change and that hardware is almost immutable. However, when one compares the typical timescales for changing hardware with the length of time it takes to create new legislation, one must expect the Chinese to have moved on to a new generation of blocking hardware long before laws to compel vendors to circumvent the firewall could reach the statute book.[48] One might reasonably expect new hardware to take account of our work and be immune to the ignoring of resets, so we conclude that legislation (vendor compulsion) is not going to be a practical way to influence events unless that legislation can be cast in exceedingly wide-ranging terms, taking little or no account of the actual technical mechanisms that must be circumvented. This leaves encouragement – getting the vendors to care less about the Chinese Government and more about everyone else – as the most realistic way forward.

VIII. CONCLUSIONS

We have demonstrated that the "Great Firewall of China" relies on inspecting packets for specific content. When filtering rules are triggered, forged reset packets are sent to each endpoint of the TCP connection. However, the genuine packets traverse the firewall unchanged and by ignoring the resets, traffic can continue unhindered. Although further connections to the same destination are also blocked if closely related port numbers are used, ignoring resets will permit unhindered access.

This result will be of considerable significance to the Chinese authorities, who will presumably wish to strengthen their systems to fix the holes in their firewall although, as we have noted, this may not be especially easy to achieve.[49]

---

[48] "The Bill has had the longest gestation period of almost any Bill in recent years. The Scott report, which gave rise to the Bill, was published in February 1996, five and a half years ago. The Conservative Government accepted the report's recommendations and immediately issued a consultation document. The Labour party's manifesto in 1997 gave a firm pledge to take action. That was followed by the publication of the White Paper in 1998. Yet after that the Government sat on their hands, so it has taken three years for the Bill finally to be introduced." *Hansard Parliamentary Debates*, Commons, 6th ser., vol. 374 (2001), col. 457.

[49] The experiments described in this paper were performed in Spring 2006 and an initial version of this paper appeared in June 2006 at the Privacy Enhancing Technologies Workshop.

However, the result may be of less significance to Chinese residents who wish to access content unhindered because their activity can still be logged and investigated. Only if the ignoring of reset packets becomes commonplace will residents be able to claim that their firewall evasion was inadvertent. This is not entirely far-fetched because validating TCP resets to see if they have been forged is a reasonable security precaution against third-party attack for TCP/IP stack vendors to be taking.

We have also shown that a side-effect of the blocking is the potential for a denial-of-service attack, albeit one that can only be used to attack particular pairs of endpoints. It is perhaps unsurprising that a blocking mechanism can be used to block things; nevertheless, without adding significant amounts of "state" to the firewall we do not see an easy way to prevent attacks.

The results we have demonstrated are also relevant to other countries, institutions and enterprises that use similar reset mechanisms to protect their interests. They should carefully note that the blocking entirely relies upon the acquiescence of those who are being blocked. Countries smaller than China may run a greater risk of denial-of-service, because they are likely to have fewer endpoints within their borders, so the firewall may not run out of resources to store details of blocked connections before the effect becomes significant.

Further experiments by another research group in Spring 2007 have uncovered some minor changes in the detail of how the firewall works, but that the reset mechanism is essentially unchanged. However, their measurements indicate that resets are now being generated within the Chinese Internet, and not at border routers, and they found that the blocking appears to be far more intermittent during busy periods than we observed a year earlier. Their research methods have also allowed them to publish an initial mapping of the range of topics that are being filtered. Jedidiah R. Crandall and others, "ConceptDoppler."